REPORT DOCUMENTATION PAGE	Form Approved OMB No. 0704-0188

1. REPORT DATE ***(DD-MM-YYYY)***	**2. REPORT TYPE**	**3. DATES COVERED** *(From – To)*
13-02-2006	Final Report	5 November 2004 - 05-Nov-05

4. TITLE AND SUBTITLE Distributed Compression In Camera Sensor Networks	**5a. CONTRACT NUMBER** FA8655-04-1-3061
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) Professor Pier Luigi Dragotti	**5d. PROJECT NUMBER**
	5d. TASK NUMBER
	5e. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Imperial College London Exhibition Road London SW7 2AZ United Kingdom	**8. PERFORMING ORGANIZATION REPORT NUMBER** N/A
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) EOARD PSC 821 BOX 14 FPO 09421-0014	**10. SPONSOR/MONITOR'S ACRONYM(S)**
	11. SPONSOR/MONITOR'S REPORT NUMBER(S) Grant 04-3061

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This report results from a contract tasking Imperial College London as follows: This effort will address a distributed compression problem using information theoretic methods originating in the work of Slepian and Wolf for lossless compression and extended by Wyner and Ziv to the case of lossy compression of continuous-valued sources. The theories developed in these papers are non-constructive and rely on asymptotic random coding arguments. Constructive designs of encoders for the distributed compression problem based on channel codes have been subsequently proposed iwith applicability to sensor networks. However, in a realistic context the statistics of the source are not known a-priori and channels codes such as turbo or trellis codes might be too complicated in this context. This effort will make use of the correlation structure of the data given by the plenoptic function n the case of multi-camera systems. In many cases the structure of the plenoptic function can be estimated without requiring inter-sensor communications, but by using some a-priori global geometrical information. Once the structure of the plenoptic function has been predicted, it is possible to develop specific distributed compression algorithms that do not require the use of complicated channel codes. This effort will develop techniques to predict the structure of the plenoptic function and develop very simple and efficient distributed compression algorithms derived from a design of a new fully distributed image compression scheme for multi-view images. The algorithm will be implemented in Matlab or C and will operate on some sets of pre-selected multi-view images. Commented MatLab psuedo-code and or C code will be provided with any executables demonstrating the algorithms.

15. SUBJECT TERMS

EOARD, Sensor Technology, Image Fusion, Ad-hoc Networks

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18, NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**	UL	22	PAUL LOSIEWICZ, Ph. D.
UNCLAS	UNCLAS	UNCLAS			**19b. TELEPHONE NUMBER** *(Include area code)* +44 20 7514 4474

Distributed Compression in Camera Sensor Networks*

Final Report

Principal Investigator: Pier Luigi Dragotti
Communications and Signal Processing Research Group
Department of Electrical and Electronic Engineering
Imperial College London

1 Introduction and Motivation

Recent advances in sensor network technology are having a profound impact on the way in which we sense, process and transport signals of interest. A sensor network consists of a large number of sensor nodes that are densely deployed, either inside or close to a phenomenon of interest. Each sensor node is an independent, low-power, smart device with sensing, processing and wireless communication capabilities. The range of applications for sensor networks is extraordinary wide and covers numbers of different areas such as health, military and home. An excellent survey on sensor networks can be found in [4].

In this project, we focus on camera sensor networks, that is, we assume that each sensor is equipped with a digital camera and transmits the acquired visual information to a common central receiver. The sensors are all observing a certain scene from different viewing positions. The images acquired by different sensors are therefore highly correlated. If the sensors were allowed to communicate with each other, it would be easy to exploit this correlation in full and transmit only the necessary information to the receiver. However, such a collaboration is usually not feasible since it would require a complex inter-sensor communication system that would consume most of the sensors' power. It is therefore necessary to develop separate compression algorithms that would still be able to exploit the correlation without requiring any cooperation amongst the sensors.

The distributed compression problem has its information theoretic origins in two papers by Slepian and Wolf [25], and by Wyner and Ziv [28], which deal with the lossless and lossy compression cases respectively. The theories developed in these papers, however, are non-constructive and rely on asymptotic random coding arguments. The first constructive design of encoders for the distributed compression problem was presented in [16] (see also [17]) and is based on the use of trellis codes. Other more sophisticated channel codes have been subsequentely presented in [1, 7]. Clearly, these distributed compression techniques can be used in camera sensor networks. However, in a realistic context the statistics of the source are not known a-priori and channels codes such as turbo or trellis codes might be too complicated in this context.

The novelty of our approach is that we explicitly use the knowledge of the spatio-temporal structure of the visual data, which is well described by the plenoptic function [3], to design our

*Notice that this work has led so far to two conference papers ICIP'05 [8] and DCC'06 [9].

compression algorithms. The key insight here is that, in many situations, the spatio-temporal structure of the plenoptic function is particularly constrained and we exploit all the available a-priori geometric knowledge to facilitate the understanding of such constraints. In particular, camera locations are usually known with a certain precision (e.g., they might be localized with a GPS system) and the visual scene of interest might be well localized in space (e.g., assume all the cameras are pointing to the same region). These geometric elements, in particular the second one, can be used to develop new more efficient distributed compression schemes.

The report is organized as follows: in the next section, we introduce the problem of distributed source coding (DSC) and review the theoretical foundations of DSC. Moreover, some practical coding schemes are reviewed and some new applications based on DSC are highlighted. In Section 3, we present our distributed compression approach for camera sensor networks in the case of a simplified scenario and lossless compression. First, the plenoptic function is introduced, then our coding scheme is presented in detail. In particular, we show that our approach allows for a flexible allocation of the bit-rates amongst the encoders, and we propose a solution to the problem of occlusions. In Section 4, we focus on the case of lossy compression and more realistic multi-view images. Finally, we conclude in Section 5.

2 Distributed Source Coding

2.1 Theoretical Background

Consider a communication system where two discrete correlated sources X and Y are to be encoded at rates R_1 and R_2 respectively, and transmitted to a central receiver. If it were possible to perform the coding jointly, a rate $R_1 + R_2 \geq H(X,Y)$ would be sufficient to perform noiseless coding. Now assume that these two sources are physically separated and cannot communicate with one another, Slepian and Wolf [25] showed that lossless compression of X and Y is still achievable if $R_1 \geq H(X|Y)$, $R_2 \geq H(Y|X)$ and $R_1 + R_2 \geq H(X,Y)$. This means that there is no loss in terms of overall rate even though the encoders are separated (see Figure 1).

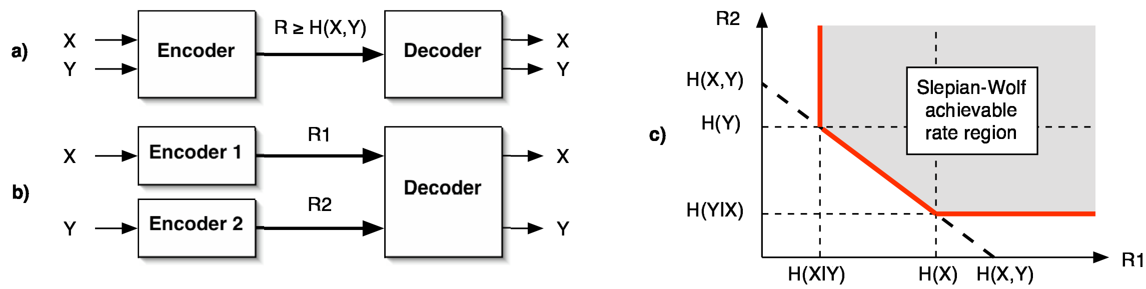

Figure 1: (a) Joint source coding. (b) Distributed source coding. The Slepian-Wolf theorem (1973) states that a combined rate of $H(X,Y)$ remains sufficient even if the correlated signals are encoded separately. (c) The achievable rate region is given by: $R_1 \geq H(X|Y)$, $R_2 \geq H(Y|X)$ and $R_1 + R_2 \geq H(X,Y)$.

The proof of the achievability in the Slepian-Wolf theorem is based on random binning. *Bin-*

ning, which refers to the partitioning of the space of all possible outcomes of a random source into different subsets, is a key concept in distributed source coding. The proof is asymptotical and non-constructive, and no practical coding approach was proposed at that time. An extension of the Slepian-Wolf result to the lossy case (with continuous sources) was proposed by Wyner and Ziv in [28]. They addressed a particular case of DSC known as *source coding with side information at the receiver* (see Figure 2). Namely, they gave a rate-distortion function $R^*_{WZ}(D)$ for the problem of encoding one source X, guarantying an average fidelity of $E\{d(X,\hat{X})\} \leq D$, assuming that the other source (playing the role of side information) is available losslessly at the decoder, but not at the encoder. In particular, they showed that, although Wyner-Ziv coding usually suffers rate loss compared to the case where the side information is available at both the encoder and decoder, there is no performance loss if the two correlated sources X and Y are jointly Gaussian and MSE is used as the distortion metric. This particular result makes Wyner-Ziv coding of great interest for practical applications, since images and video sources are sometimes modeled as jointly Gaussian (after mean substraction).

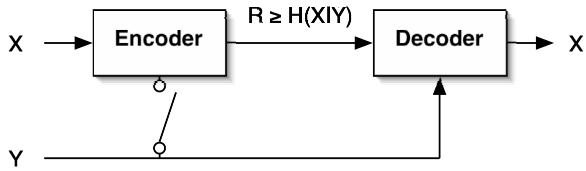

Figure 2: Lossy compression of X with side information Y. Wyner and Ziv showed that if X and Y are jointly Gaussian and MSE is used as the distortion metric, there is no performance loss whether the side information Y is available at the encoder or not, as long as it is available at the decoder.

Slepian-Wolf and Wyner-Ziv coding are source coding problems. However, a strong link to channel coding exists, since the practical binning schemes used at the encoders are usually based on linear channel codes and their coset codes. The next section presents the general idea behind the design of practical coders based on channel coding principles.

2.2 Practical Coders

In DISCUS [16], Pradhan and Ramchandran proposed for the first time a practical coding technique for DSC inspired by channel coding techniques. In fact, the link between distributed source coding and channel coding had already been made at the time by Wyner [27], but nobody had used it to design practical coders. In order to give the correct intuition behind the DISCUS approach, we start with a simple example: Assume x and y are two uniformly distributed 3-bit sequences that are correlated such that their Hamming distance is at most one (i.e., for any realization of x, y is either equal to x or just differs at one bit's position). Therefore, given a certain x, we know that the corresponding y belongs to an equiprobable set of four codewords. The following entropies can thus be given: $H(x) = H(y) = 3$ bits, $H(x|y) = H(y|x) = 2$ bits and $H(x,y) = H(x) + H(y|x) = 5$ bits. Therefore, only 5 bits are necessary to jointly losslessly encode x and y. For instance, one can code x and y jointly by sending one of them completely (3 bits) along with the information representing their difference (2 bits).

According to Slepian and Wolf, it is possible to achieve the same coding efficiency using two independent encoders. The solution consists in grouping the different codewords into bins. Assume that y is transmitted completely to the decoder (using 3 bits), and consider the following set of bins containing all the possible outcomes for x: $\text{bin}_0 = \{000, 111\}$, $\text{bin}_1 = \{001, 110\}$, $\text{bin}_2 = \{010, 101\}$ and $\text{bin}_3 = \{100, 011\}$. Note that the codewords have been placed into the bins such that the distance between the members of a given bin is maximal (3 in this case). Now, instead of transmitting x perfectly to the decoder (3 bits), only the index of the bin that x belongs to is transmitted (2 bits). On receiving this information, the decoder can retrieve the two possible candidates for x. Finally, since their distance to each other is three, only one of them can satisfy the correlation with y given by: $d_H(x, y) \leq 1$. By observing y, the decoder can therefore retrieve the right value of x.

This intuitive example can be generalized using linear channel codes. Assume that x and y are two uniformly distributed n-bit sequences that are correlated such that their Hamming distance is at most m, i.e. $d_H(x, y) \leq m$. Consider an (n, k) channel code $\mathcal{C}$, given by its parity check matrix $\mathbf{H}$, that can correct up to $M \geq m$ errors per n-bit codeword. We call *coset number i* the set $\{x_j\}_{j=1}^{2^k}$ of all n-bit codewords that have a syndrome equal to i (i.e., $\mathbf{H}x_j^T = i$). The code $\mathcal{C}$ generates thus 2^{n-k} cosets having 2^k members each. Moreover, any pair of codewords belonging to the same coset have a Hamming distance larger than $2M$. Similarly to our previous example, the distributed coding strategy operates as follows: y is sent perfectly from the second encoder (n bits). The first encoder only transmit the syndrome $s_x = \mathbf{H}x^T$ ($n - k$ bits). At the decoder, the original x can be recovered as the only member of coset s_x satisfying the correlation ($d_H(x, y) \leq m$) with the received y. This distributed encoding approach requires thus only $2n - k$ bits to transmit x and y losslessly.

Practical designs based on advanced channel codes such as Turbo and LDPC have been proposed in several papers (see [7, 1, 14] for example). They all propose practical coding approaches that can closely approach the theoretical bounds for different kind of correlation models. Nevertheless, most of these approaches focus on the asymmetric scenario, also known as compression with side information at the decoder. Approaches allowing to cover the entire Slepian-Wolf achievable rate region have recently been proposed in [26, 20, 6].

The link between distributed source coding and channel coding is highlighted in Figure 3. In channel coding, a redundant codeword x is generated by adding parity bits to the original information block c to be transmitted, such that, after x is sent through the noisy channel, the corrupted output y still contains enough information to perfectly recover c. In other terms, the idea is to determine a set of x's, such that, when any of them is sent through the noisy channel, the received y is still closer to the original x than to any other member of this set, so that it is possible to retrieve x from y. An appropriate code is therefore chosen based on the joint distribution $p(x, y)$. In distributed source coding, x and y represent the two correlated sources to be transmitted. Assuming that y has been transmitted to the decoder, only the syndrome s of x needs to be transmitted from the first source. At the decoder, the set of all x's having syndrome s is retrieved and the only one satisfying the correlation with y is retrieved as the original x.

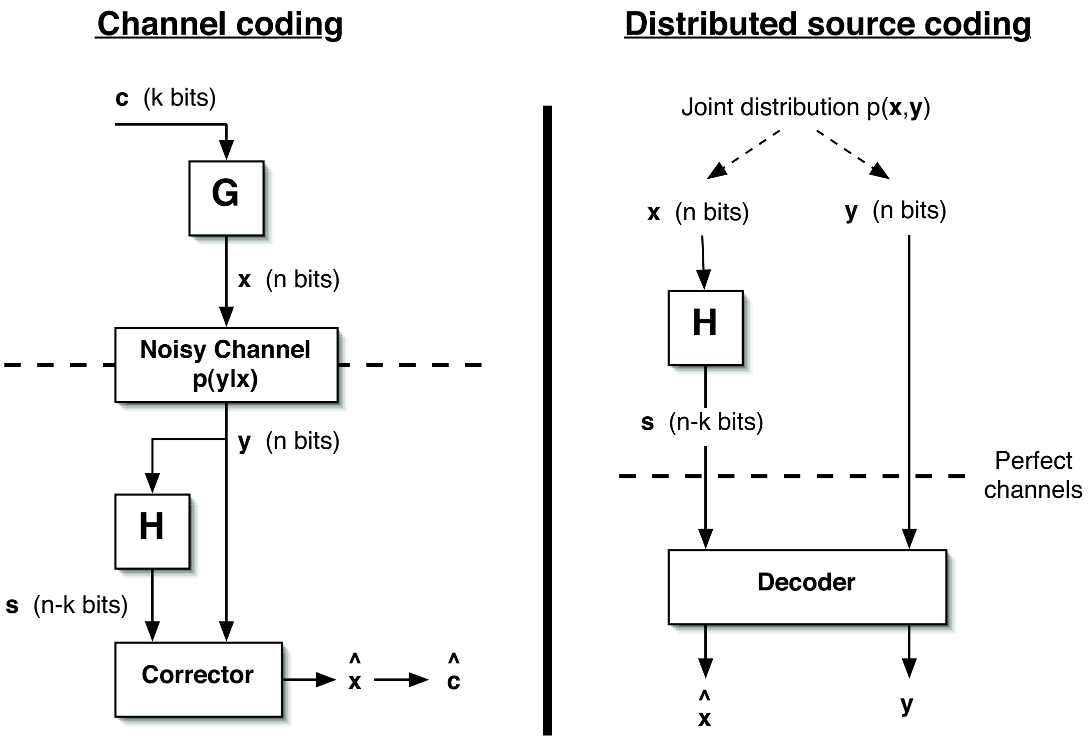

Figure 3: Channel coding vs. Distributed source coding. In channel coding, the syndrome of y is used at the decoder to determine the error pattern. Then the original x is recovered by correcting y. In distributed source coding, the syndrome of x is transmitted to the decoder. Knowing y and the syndrome of x, the decoder can thus retrieve the difference pattern between x and y and then reconstruct the original x.

2.3 New Applications of Wyner-Ziv

2.3.1 Distributed Video Coding

In video coding standards such as MPEG or the more recent H.264, the encoder usually tries to exploit the statistics of the source signal in order to remove, not only spatial, but also temporal redundancies. This is usually achieved using motion-compensated predictive encoding, where each video frame is encoded using a prediction based on previously encoded frames. The prediction can be seen as a sort of side information and, in this case, is available at both the encoder and the decoder.

The idea of distributed video coding is to employ DSC approaches in order to allow for an independent encoding of the different frames at the encoder, while letting to the joint decoder the burden of exploiting the temporal dependencies. In other terms, each video frame is encoded independently knowing that some side information will be available at the decoder (the side information can typically be a prediction based on previously decoded frames).

The first very interesting aspect of distributed video coding is that it considerably reduces the complexity of the video encoder by shifting all the complex interframe processing tasks to the decoder. This property can be of great interest for power/processing limited systems such as wireless camera sensors that have to compress and send video to a fixed base station in a power-efficient way. Here, it is assumed that the receiver has the ability to run a more complex decoder. In the case where the receiver of the compressed video signal is another complexity-constrained device, a solution using a more powerful video transcoder somewhere on the network can be used (see Figure 4).

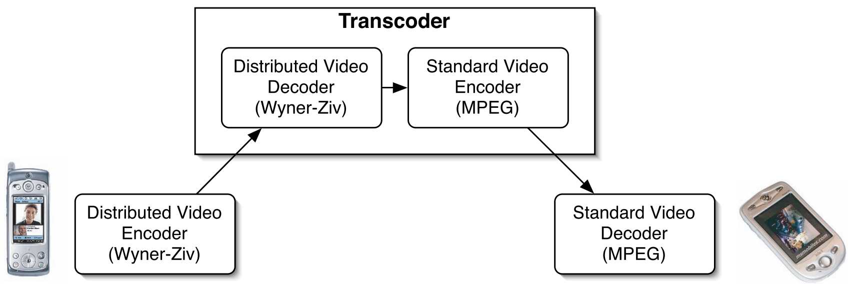

Figure 4: Transcoding architecture for wireless video. This method allows for a low-complexity encoder (Wyner-Ziv encoder) and decoder (MPEG decoder) at both wireless devices. However, this architecture relies on the use of a complex transcoder somewhere on the network.

Another strong advantage of distributed video coding is that it is naturally robust to the problem of drift between encoder and decoder. The drift problem is due to prediction mismatch that can happen due to channel loss and usually creates visual artifacts that propagates until the next intra-coded frame is received. This built-in robustness is due to the fact that the encoding is not based on a specific prediction, but only assumes that a relatively good predictor will be available at the decoder. Therefore, slightly different predictors can lead to a correct decoding. This particular property highlights the fact that Wyner-Ziv coding can actually be seen as a source-channel coding problem (see Section 2.3.3).

The first video coding approach based on distributed compression principles was proposed in [18], and is known as *PRISM*. We urge the reader to refer to this original work to obtain more information about their specific coding architecture. More recently, another video coding approach based on DSC was proposed in [22], where the authors clearly focused on the robustness introduced by the use of Wyner-Ziv coding. Finally, a third similar approach can be found in [10].

Although all these approaches are extremely promising, they are still not as efficient as standard video coders in terms of rate-distortion performance. The gap is mainly due to the fact that distributed source coding techniques usually rely on the fact that the correlation structure is known a-priori. It is therefore only with the knowledge of this correlation that optimal codes can be designed. The estimation of this correlation has proven, however, to be extremely difficult.

2.3.2 Multi-Camera Arrays

Compression techniques for multi-view images have attracted a deep interest during the last decade. This is partly due to the introduction of several new $3D$ rendering techniques such as image-based rendering (IBR) and lightfield rendering (LFR) that represent real-world $3D$ scenes using a set of images obtained from fixed viewpoint cameras. The amount of raw data acquired by practical systems can be extraordinary large and typically consists of hundreds of pictures. Due to the spatial proximity of the different cameras, an extremely large amount of redundant information is present in the acquired data. Compression is therefore highly needed.

In order to exploit the correlation between the different views, a joint encoder should be employed. However, this would require that all the cameras first transmit their raw data to a common receiver that would have to store it and then perform the joint compression. This would clearly use a tremendous amount of transmission resources and storage space, and might not be feasible in some practical settings. For these reasons, it would be preferable to compress the images directly at the cameras using distributed compression techniques. The main advantages of such an approach is that it would only require a low-complexity encoder at each camera, and would considerably reduce the overall amount of transmission necessary from the cameras to the central decoder. Moreover, the compressed data could be directly stored at the receiver using optimal memory space. Nevertheless, in this case the decoder is assumed to be more sophisticated in order to handle the high-complexity joint decoding of the views, when necessary.

Several approaches inspired by distributed video coding have been proposed recently [30, 12, 2]. The basic idea is to see each different view as a frame of a video sequence and apply a Wyner-Ziv video coding approach to them. Nevertheless, these approaches suffer from several drawbacks: First, they require that some cameras transmit their full information (to provide side information to the receiver) while others only transmit partial information. This makes them clearly asymmetric, which can be a problem for some practical applications. Second, while the correlation between successive video frames can be difficult to estimate, basic multi-view geometry could be used when dealing with multi-camera systems. However, none of these approaches takes advantage of this information so as to improve the performance of their encoders.

2.3.3 Joint Source-Channel Coding

As stated in an excellent review on distributed source coding by Xiong et al. [29]; "Wyner-Ziv coding is, in a nutshell, a source-channel coding problem". This property of Wyner-Ziv was highlighted in Section 2.3.1, where we addressed the fact that distributed video coding presents a natural robustness to the problem of drift. In fact, Wyner-Ziv coding can be thought of as a channel coding technique that is used to correct the "errors" between the source to be coded and the side information. If we assume that the relationship between the source and the side information is modeled by a "virtual" correlation channel. Then, if a good channel code for this "virtual" channel can be found, it would clearly provide us with a good Wyner-Ziv code through the associated coset codes.

In case of transmission over a non-perfect channel, it seems quite intuitive that the use of a stronger Wyner-Ziv code could not only compensate for the discrepancies between the source and the side information, but also correct errors due to the unreliable transmission of the source sequence. Several papers addressing this particular property of distributed source coding have recently been published [15, 21].

Finally, Wyner-Ziv coding is also strongly related to systematic lossy source-channel coding [23], where an encoded version of the source signal is sent over a digital channel to serve as enhancement information to a noisy version of the source signal received through an analog channel. Here, the noisy version of the source signal plays the role of side information for decoding the information received from the digital channel. A detailed description of video coding based on systematic lossy source-channel coding can be found in [19].

3 Distributed Compression in Camera Sensor Networks

Distributed compression schemes usually rely on the assumption that the correlation of the source is known a-priori. In this section, we show how it is possible to estimate the correlation structure in the visual information acquired by a multi-camera system by using some simple geometrical constraints, and present a coding approach that can exploit this correlation in order to reduce the overall transmission bit-rate from the camera sensors to the common central receiver. The coding scheme we propose allows for a flexible distribution of the bit-rates amongst the encoders and is optimal in many cases. Our technique can intuitively be extended to the general case of binary sources, and can also be made resilient to a fixed number of visual occlusions.

3.1 The Plenoptic Function

The plenoptic function was first introduced by Adelson and Bergen in 1991 [3]. It corresponds to the function representing the intensity and chromaticity of the light observed from every position and direction in the 3D space, and can therefore be parameterized as a 7D function: $P_7 = P(\theta, \phi, \lambda, t, V_x, V_y, V_z)$. This function represents thus all the visual information available from any viewing position around a scene of interest. Hence, image-based rendering (IBR) techniques can be thought of as methods that try to reconstruct the continuous plenoptic function from a finite set of views. Once the plenoptic function has been reconstructed, it is then straightforward to generate any view of the scene by setting the appropriate parameters. The high dimensionality of this function makes it, however, extremely impractical. By fixing the time t and the wavelength λ, and assuming that the whole scene of interest is contained in a convex hull, the plenoptic function can be reduced to a 4-D function. Several methods for representing this 4-D function and for reconstructing it from sample images have been proposed [11, 13]. The parameterization of this 4-D function is usually done using two parallel planes: the focal plane (or camera plane) and the retinal plane (or image plane). A ray of light is therefore parameterized by its intersection with these two planes. The coordinates in the focal plane (s,t) gives the position of the pinhole camera, while the coordinates in the retinal plane (u,v) gives the point in the corresponding image.

Epipolar plane images (EPI) are usually used to represent the redundancy in the plenoptic function such that it can be exploited easily. The idea is to restrict our attention to a 2-D subspace of the plenoptic function. For example, the (v,t) plane is usually used to represent the epipolar geometry of a scene, assuming that the pinhole cameras are placed on a horizontal line (see Figure 6).

3.2 Our Camera Sensor Network Configuration

A camera sensor network is able to acquire a finite number of different views of a scene at any given time and can thus be seen as a sampling device for the plenoptic function. We choose the following scenario for our work: Assume that we have N cameras placed on a horizontal line. Let α be the distance between two consecutive cameras, and assume that they are all looking in the same direction (perpendicular to the line of cameras). Assume that the observed scene is composed of simple objects such as uniformly colored polygons parallel to the image plane and with depths bound between the two values z_{min} and z_{max} as shown in Figure 5. According

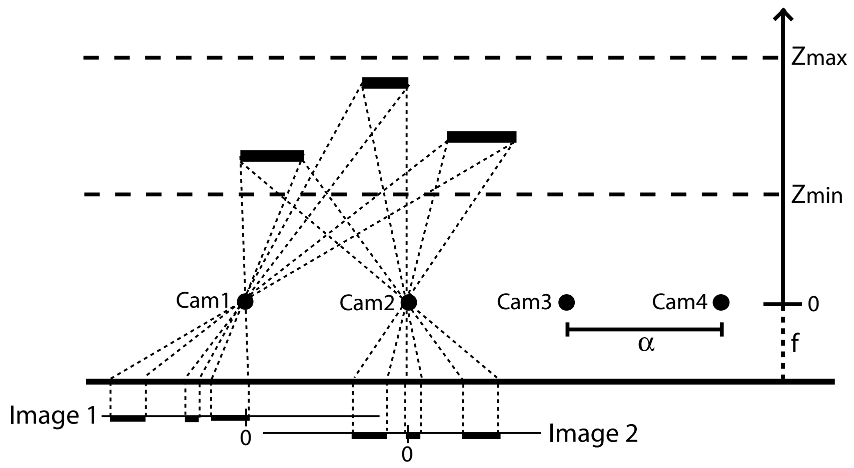

Figure 5: Our camera sensor network configuration.

to the epipolar geometry principles, which are directly related to the structure of the plenoptic function (see Figure 6), we know that the difference between the positions of a specific object on the images obtained from two consecutive cameras will be equal to $\Delta = \frac{\alpha f}{z}$, where z is the depth of the object and f is the focal length of the cameras. This disparity Δ depends only on the distance z of the point from the focal plane. If we know a-priori that there is a finite depth of field, that is $z \in [z_{min}, z_{max}]$, then there is a finite range of disparities to be coded, irrespective of how complicated the scene is. This key insight can be used to develop new distributed compression algorithms as we show in the next section.

Notice that a similar insight has been previously used by Chai et al. to develop new schemes to sample the plenoptic function [5].

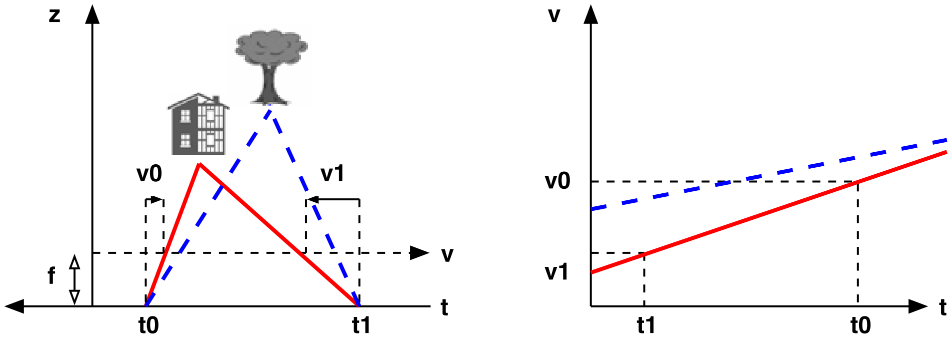

Figure 6: 2D plenoptic function of two points. The t-axis corresponds to the camera position and v corresponds to the relative positions on the corresponding image. A point of the scene is therefore represented by a line whose slope is directly related to the point's depth (z-axis). The difference between the positions of a given point on two different images thus satisfies the relation $(v - v') = \frac{f(t-t')}{z}$, where z is the point's depth and f is the focal length of the cameras.

3.3 Our Coding Approach

In this section, we propose a distributed coding scheme for the configuration presented in Figure 5 with two cameras. Since both encoders have some knowledge about the geometry of the scene, the correlation structure of the two sources can be easily retrieved. We then show that our coding technique can be used with any pair of bit-rates contained in the achievable rate region defined by Slepian and Wolf.

3.3.1 Asymmetric encoding

Let X and Y be the horizontal positions of a specific object on the images obtained from two consecutive cameras. Assume the image width is made of 2^R pixels. Due to the epipolar geometry and the information we have about the scene, that is $(\alpha, f, z_{min}, z_{max})$, we know that $Y \in [X + \frac{\alpha f}{z_{max}}, X + \frac{\alpha f}{z_{min}}]$ for a specific X. Encoding X and Y independently would require a total of $H(X) + H(Y)$ bits. However, using a *coset-like* approach, we can transmit X losslessly and modulo encode Y as $Y' = Y \bmod \lceil \alpha f (\frac{1}{z_{min}} - \frac{1}{z_{max}}) \rceil$. By observing X and Y', the receiver will then retrieve the correct Y such that $Y \in [X + \frac{\alpha f}{z_{max}}, X + \frac{\alpha f}{z_{min}}]$. The overall transmission rate is therefore decreased to $H(X) + H(Y')$ bits. If we assume that the difference between X and Y is uniformly distributed in $[\frac{\alpha f}{z_{max}}, \frac{\alpha f}{z_{min}}]$, we can claim that $H(Y') = H(Y|X)$. We can see that our coding scheme uses $H(X) + H(Y') = H(X) + H(Y|X) = H(X, Y)$ bits and is therefore optimal.

This simple distributed coding technique is very powerful since it takes full advantage of the geometrical information to minimize the global transmission bit-rate. However, its asymmetric repartition of the bit-rates may be problematic for some practical applications. In the following, we will show that our coding approach can be extended in a way such that any pair of bit-rates satisfying the Slepian and Wolf conditions can be used.

3.3.2 Flexible distribution of the bit-rates

Looking at the following relation: $H(X,Y) = H(X|Y) + H(Y|X) + I(X,Y)$, we can see that the minimum information that must be sent from the source X corresponds to the conditional entropy $H(X|Y)$. Similarly, the information corresponding to $H(Y|X)$ must be sent from the source Y. The remaining information required at the receiver in order to recover the values of X and Y perfectly is related to the mutual information $I(X,Y)$ and is by definition available at both sources. This information can therefore be obtained partially from both sources in order to balance the transmission rates.

We know that the correlation structure between the two sources is such that Y belongs to $[X + \frac{\alpha f}{z_{max}}, X + \frac{\alpha f}{z_{min}}]$ for a given X. Let $\widetilde{Y}$ be defined as $\widetilde{Y} = Y - \lceil \frac{\alpha f}{z_{max}} \rceil$. This implies that the difference $(\widetilde{Y} - X)$ is contained in $\{0, 1, \ldots, \delta\}$, where $\delta = \lceil \alpha f(\frac{1}{z_{min}} - \frac{1}{z_{max}}) \rceil$. Looking at the binary representations of X and $\widetilde{Y}$, we can say that the difference between them can be computed using only their last R_{min} bits where $R_{min} = \lceil \log_2(\delta + 1) \rceil$. Let X_1 and $\widetilde{Y}_1$ correspond to the last R_{min} bits of X and $\widetilde{Y}$ respectively. Let $X_2 = (X \gg R_{min})$ and $\widetilde{Y}_2 = (\widetilde{Y} \gg R_{min})$, where the "$\gg$" operator corresponds to a binary shift to the right. We can thus say that $\widetilde{Y}_2 = X_2$ if $\widetilde{Y}_1 \geq X_1$ and that $\widetilde{Y}_2 = X_2 + 1$ if $\widetilde{Y}_1 < X_1$. As presented in Figure 7, our coding strategy consists in sending X_1 and $\widetilde{Y}_1$ from the sources X and Y respectively and then, sending only a subset of the bits for X_2 and only the complementary one for $\widetilde{Y}_2$. At the receiver, X_1 and $\widetilde{Y}_1$ are then

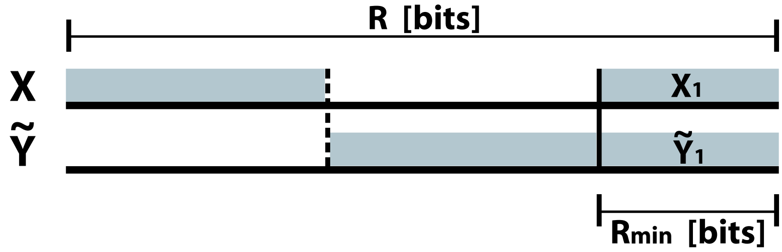

Figure 7: Binary representation of the two correlated sources. The last R_{min} bits are sent from the two sources but only complementary subsets of the first $(R - R_{min})$ bits are necessary at the receiver for a perfect reconstruction of X and Y.

compared to determine if $\widetilde{Y}_2 = X_2$ or if $\widetilde{Y}_2 = X_2 + 1$. Knowing this relation and their partial binary representations, the decoder can now perfectly recover the values of X and $\widetilde{Y}$.

Assume that z_{min} and z_{max} are such that $(\delta+1)$ is a power of 2. Since we assume that $(\widetilde{Y} - X)$ is uniformly distributed, we can state that $H(\widetilde{Y} - X) = H(X|Y) = H(Y|X) = R_{min}$. Let $S(X_2)$ be a subset of the $R - R_{min}$ bits of X_2 and let $\bar{S}(\widetilde{Y}_2)$ corresponds to the complementary subset of $\widetilde{Y}_2$. If we assume now that X is uniformly distributed in $\{0, 1, \ldots, 2^R - 1\}$, we can say that $H(S(X_2)) + H(\bar{S}(\widetilde{Y}_2)) = H(S(X_2), \bar{S}(\widetilde{Y}_2)) = I(X,Y)$. The total rate necessary for our scheme corresponds to $I(X,Y) + 2R_{min} = H(X,Y)$ and is therefore optimal. We can now summarize our results into the following proposition:

Proposition 1 *Consider the configuration presented in Figure 5 with two cameras, and assume that no occlusion happens in the two corresponding views. The following distributed coding strategy is sufficient to allow for a perfect reconstruction of these two views at the decoder. For each object's* *position:*

- *Send the last R_{min} bits from both sources, with $R_{min} = \lceil \log_2(\delta + 1) \rceil$ and $\delta = \lceil \alpha f(\frac{1}{z_{min}} - \frac{1}{z_{max}}) \rceil$.*
- *Send complementary subsets for the first $(R - R_{min})$ bits.*

If we assume that X and $(Y - X)$ are uniformly distributed and that $\delta = 2^{R_{min}} - 1$, this coding strategy achieves the Slepian-Wolf bounds and is therefore optimal.

3.3.3 The problem of occlusions

In order to reconstruct the position of an object for any virtual camera position, we need to know its correct position in at least two different views. Using the epipolar geometry principles, we can then easily retrieve its absolute position and depth. Unfortunately, a specific object may not be visible from certain view points since it might be hidden behind another object or might be out of field. Nevertheless, using a configuration with more cameras will make it more likely for any object to be visible in at least two views.

Assume we have three cameras in a configuration similar to the one presented in Figure 5 and that each object of the scene can be occluded in at most one of these three views. Our goal is to design a distributed coding scheme for these three correlated sources such that the information provided by any pair of these sources is sufficient to allow for a perfect reconstruction at the receiver. Let X, Y and Z be the horizontal positions of a specific object on the images obtained from camera 1, 2 and 3 respectively. We know that Y belongs to $[X + \frac{\alpha f}{z_{max}}, X + \frac{\alpha f}{z_{min}}]$ and Z belongs to $[X + 2\frac{\alpha f}{z_{max}}, X + 2\frac{\alpha f}{z_{min}}]$ for a given X. Moreover, we know that any of these variables is deterministic given the two others and follows the relation $Z = 2Y - X$. Let $\widetilde{X}$ and $\widetilde{Z}$ be defined as $\widetilde{X} = X + \frac{\alpha f}{z_{mean}}$ and $\widetilde{Z} = Z - \frac{\alpha f}{z_{mean}}$ where z_{mean} is defined such that $\frac{1}{z_{mean}} = \frac{1}{2}(\frac{1}{z_{min}} + \frac{1}{z_{max}})$. This implies that the differences $(Y - \widetilde{X})$ and $(\widetilde{Z} - Y)$ are equal and belong to $[-\delta/2, \delta/2]$ and that the difference $(\widetilde{Z} - \widetilde{X})$ belongs to $[-\delta, \delta]$, where δ is defined as in Section 3.3.2.

Looking at the binary representation of $\widetilde{X}$, Y and $\widetilde{Z}$ (at integer precision), we can say that the difference between any pair can be retrieved using only their last R_{min} bits, where $R_{min} = \lceil \log_2(2\delta + 1) \rceil$. Let $\widetilde{X}_1$, Y_1 and $\widetilde{Z}_1$ correspond to the last R_{min} bits of $\widetilde{X}$, Y and $\widetilde{Z}$ respectively. Using a similar approach to that presented in Section 3.3.2, we know that any complementary binary subsets of $\widetilde{X}_2$, Y_2 and $\widetilde{Z}_2$ are necessary at the receiver to allow for a perfect reconstruction. Since one occlusion can happen, we have to choose the binary subsets such that any pair of these subsets contains at least one value for each of the $(R - R_{min})$ bits. A possible repartition is shown in Figure 8 (symmetric case). A transmission rate of $\frac{2}{3}r + R_{min}$ for each source is necessary in this case, where $r = R - R_{min}$.

On receiving the last R_{min} bits from only two sources, the decoder is able to retrieve the last R_{min} bits of the third one, which may be occluded. Therefore, the relationship between $\widetilde{X}_2, Y_2$ and $\widetilde{Z}_2$ can be obtained and only subsets of their binary representations are necessary for a perfect reconstruction.

3.3.4 Generalization to N cameras with M possible occlusions

We can now generalize our result to any number of cameras and occlusions with the following proposition (see Figure 9):

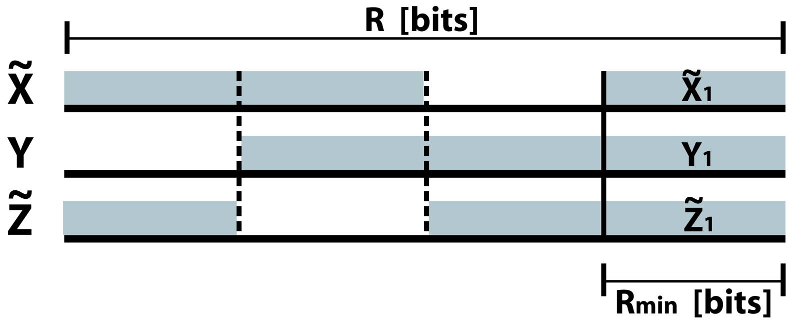

Figure 8: Binary representation of the three correlated sources. The last R_{min} bits are sent from the three sources but only subsets of the first $(R - R_{min})$ bits are necessary at the receiver for a perfect reconstruction of X, Y and Z even if one occlusion occurs.

Proposition 2 *Consider a system with N cameras as depicted in Figure 5. Assume that any object of the scene can be occluded in at most $M \leq N-2$ views. The following distributed coding strategy is sufficient to allow for a perfect reconstruction of these N views at the decoder and to interpolate any new view:*

- *Send the last R_{min} bits of the objects'positions from only the first $(M+2)$ sources, with $R_{min} = \lceil \log_2((M+1)\delta) \rceil$ and $\delta = \lceil \alpha f(\frac{1}{z_{min}} - \frac{1}{z_{max}}) \rceil$.*
- *For each of the N sources, send only a subset of its first $(R - R_{min})$ bits such that each particular bit position is sent from exactly $(M+1)$ sources.*

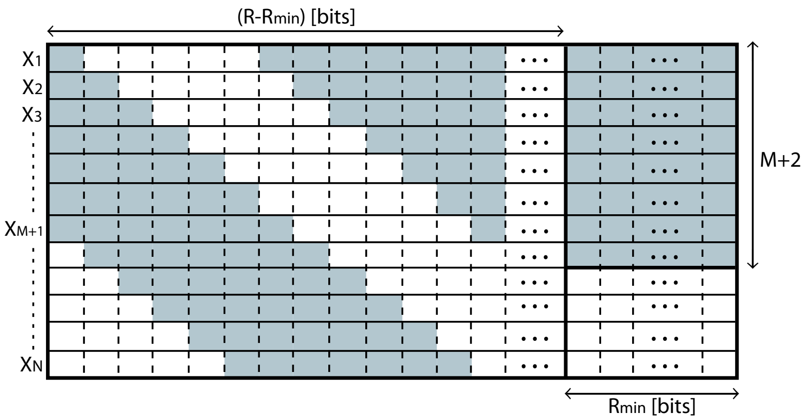

Figure 9: Binary representation of the N correlated sources. The last R_{min} bits are sent only from the $(M+2)$ first sources. Only subsets of the first $(R-R_{min})$ bits are sent from each source, such that each bit position is sent exactly from $(M+1)$ sources.

3.3.5 Simulation results

We developed a simulation to illustrate the performance of our distributed compression scheme. We created an artificial scene composed of simple objects such as polygons of different intensities placed at different depths. Our system could then generate any view of that scene for any specified camera position. In the example presented in Figure 10, we generated three views of a simple scene composed of three objects such that one of them is occluded in the second view, and another one is out of field in the third view. The three generated images have a resolution of 512×512

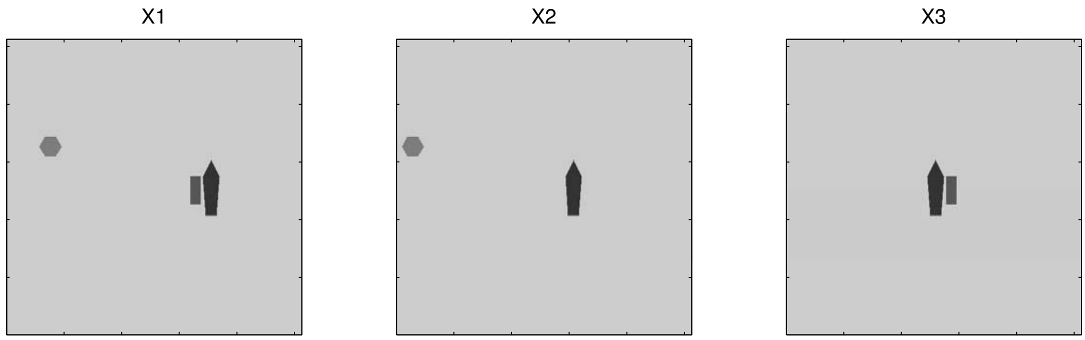

Figure 10: Three views of a simple synthetic scene obtained from three aligned and evenly spaced cameras. Note that an occlusion happens in X_2 and that an object is out of field in X_3.

pixels and are used as the inputs for the testing of our distributed compression algorithm. Each encoder applies first a simple corner detection to retrieve the vertex positions of their visible polygons. Each vertex (x, y) is represented using $2R = 2\log_2(512) = 18$ bits. Each encoder knows the relative locations of the two other cameras ($\alpha = 100$) but does not know the location of the objects on the other images. It only knows that the depths of the objects are contained in $[1.95, 5.05]$ and that $f = 1$. Depending on its depth, an object will thus be shifted from 20 to 51 pixels between two consecutive views. This means that the difference Δ on two consecutive positions can be described using $R_{min} = \log_2(51 - 19) = 5$ bits.

In order to be resilient to one occlusion, we applied the approach proposed in Section 3.3.3. The results showed that only 14 bits per vertex were necessary from each source (instead of 18) to allow for a perfect reconstruction of the scene at the receiver. When repeating the operation with three other views and assuming that no occlusion was possible, only 8 bits per vertex were necessary from each source.

4 Lossy distributed compression in Camera Sensor Network

The coding approach proposed in Section 3 can theoretically achieve the Slepian-Wolf bound and gives us a precise intuition on how distributed compression should be applied to multi-view images. However, it assumes that the location of the objects' boundaries are known a-priori, and is therefore not directly applicable to encode real multi-view images. In this section, we show how we extended the above method to the case of more realistic multi-view images. We used two different approaches. The first one based on quad-tree decomposition of the images, the second one on a distributed version of the wavelet transform.

4.1 Distributed compression using tree structured algorithms

Since the correlation model used by our distributed coding approach is related to the object's positions on the different views, we need to develop coding algorithms that can efficiently represent these positions. Our approach consists in representing the different views using a piecewise polynomial model. The main advantage of such a representation is that it is well adapted to represent real images and that it is able to precisely catch the discontinuities between the objects. Two different views can therefore be modeled using a piecewise polynomial signal where each discontinuity is shifted according to the correlation model $\Delta_i \in \{\Delta_{min}, \Delta_{max}\}$. If we assume that the scene is composed of lambertian planar surfaces and that no occlusion occurs in the different views, we can then claim that the polynomial pieces are similar for the different views.[1]

In [24], Shukla et al. presented new coding algorithms based on tree structured segmentation that achieve the correct asymptotic rate-distortion (R-D) behaviour for piecewise polynomial signals. Their method is based on a prune and join scheme that can be used for 1D (using binary trees) or for 2D (using quadtrees) signals. We highlight here the main elements of their compression algorithm for 1D signals.

Algorithm 1 Prune-Join binary tree coding algorithm [24]

1: Segmentation of the signal using a binary tree decomposition up to a tree depth J_{max}.
2: Approximation of each node of the tree by a polynomial $p(t)$ of degree $\leq P$.
3: Rate-Distortion curves generation for each node of the tree (scalar quantization of the polynomial coefficients).
4: Optimal pruning of the tree for the given operating slope $-\lambda$ according to the following Lagrangian cost based criterion: Prune the two children of a node if $(D_{C_1} + D_{C_2}) + \lambda(R_{C_1} + R_{C_2}) \geq (D_p + \lambda R_p)$.
5: Joint coding of similar neighbouring leaves according to the following Lagrangian cost based criterion: Join the two neighbours if $(D_{n_1} + \lambda R_{n_1}) + (D_{n_2} + \lambda R_{n_2}) \geq (D_{n_{Joint}} + \lambda R_{n_{Joint}})$.
6: Search for the desired R-D operating slope (update λ and go back to point 4).

Our distributed coding strategy which is based on these tree-structured algorithms can be summarized as follows. (For simplicity we focus on the 1D case).

Let $f_1(t)$ be a piecewise polynomial signal defined over $[0;T]$ consisting of $S+1$ polynomial pieces of maximum degree P each. Let $\{t_{1_i}\}_{i=1}^{S}$ be its set of the S distinct discontinuity locations. We define $f_2(t)$ as another piecewise polynomial function over $[0;T]$ having the same polynomial pieces than $f_1(t)$, but whose set of discontinuity locations $\{t_{2_i}\}_{i=1}^{S}$ is chosen such that: $\Delta_{min} \leq t_{2_i} - t_{1_i} \leq \Delta_{max}, \forall i \in \{1, \ldots, S\}$. The relationship between $f_1(t)$ and $f_2(t)$ is therefore given by the range of possible disparities $[\Delta_{min}; \Delta_{max}]$ which corresponds to the plenoptic constraints we consider in our camera sensor network scenario.

Assume that these two signals are independently encoded using the prune-join algorithm for a given distortion target. The total information necessary to describe each of them can be divided

[1]With non-lambertian surfaces, or with the presence of occlusions, the polynomial pieces can differ for the different views. Our simple correlation model should therefore be modified in this case. For the sake of simplicity, we will however only consider this simple model to present our coding approach.

in 3 parts: R_{Tree} is the number of bits necessary to code the pruned tree and is equal to the number of nodes in the tree. $R_{LeafJointCoding}$ is the number of bits necessary to code the joining information and is equal to the number of leaves in the tree. Finally, R_{Leaves} is the total number of bits necessary to code the set of polynomial approximations.

Figure 11 presents a prune-join tree decompositions of two piecewise constant signals, having the same set of amplitudes and having their sets of discontinuities satisfying our plenoptic constraints. Due to this relationship between the two signals, we can observe that the structure of the two pruned binary trees presents some similarities. Our distributed compression algorithm uses these similarities in order to transmit only the necessary information to allow for a perfect reconstruction at the decoder. It can be described as follows (asymmetric encoding):

- Send the full description of signal 1 from encoder 1. ($R_1 = R_{Tree_1} + R_{LeafJointCoding_1} + R_{Leaves_1}$)
- Send only the subtrees of signal 2 having a root node at level J_Δ along with the joining information from encoder 2, where $J_\Delta = \lceil \log_2(\frac{T}{\Delta_{max}-\Delta_{min}+1}) \rceil$. ($R_2 = (R_{Tree_2} - R_{\Delta_2}) + R_{LeafJointCoding_2}$ where R_Δ corresponds to the number of nodes in the pruned tree with a depth smaller than J_Δ.)

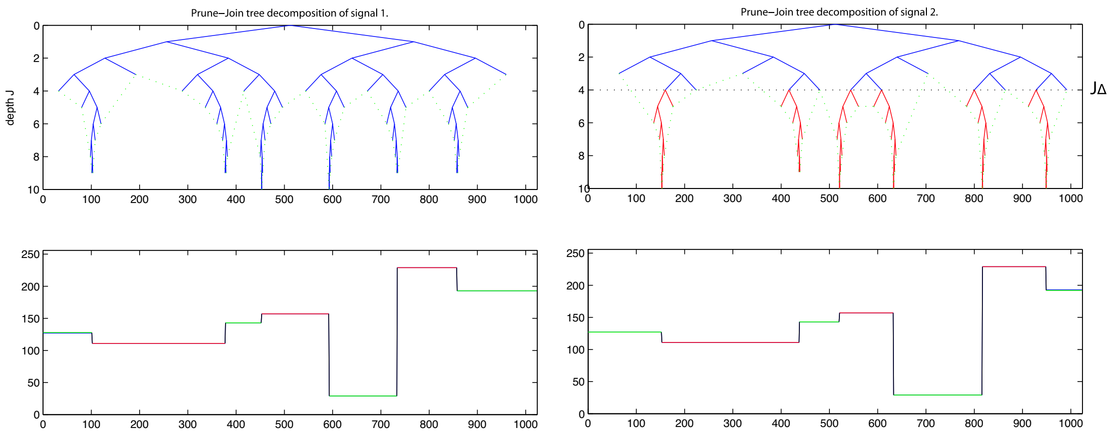

Figure 11: Prune-Join binary tree decomposition of two piecewise constant signals satisfying our correlation model.

At the decoder, the original position of the subtrees received from encoder 2 can be recovered using the side information provided by encoder 1, such that all the disparities satisfy the plenoptic constraints. The full tree can then be recovered and the second signal can thus be reconstructed using the approximations received from encoder 1.

The prune-join binary tree decomposition used in our approach has an intuitive extension to the 2D case, where the binary tree segmentation is replaced by the quad-tree segmentation and the polynomial model is replaced by a 2D geometrical model. Although our approach becomes more involved in the 2D case, the intuitions remain the same. The geometrical model used in 2D corresponds to two 2D polynomials separated by a 1D polynomial boundary. Notice that the quad-tree compression algorithm proposed in [24] outperform Jpeg2000. For this reason we are confident that its use in the multi-view context will lead to good simulation results. This is, however, part of our on-going work.

This distributed compression algorithm have been applied to a set of scan lines of real multi-view images. We present a simulation on a scan line of a pair of stereo images (Figure 12) using a piecewise linear model and a symmetric encoding strategy. The reconstructed signals (Figure 13) present a good level of accuracy for the reconstruction of the two scan-lines.

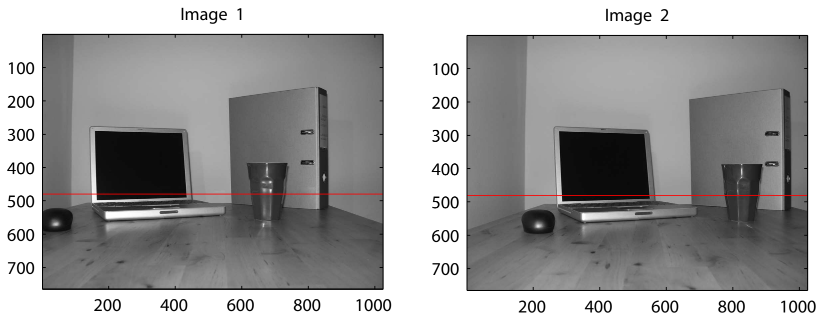

Figure 12: Stereo images of a real scene where the objects are located between a minimum and a maximum distance from the cameras.

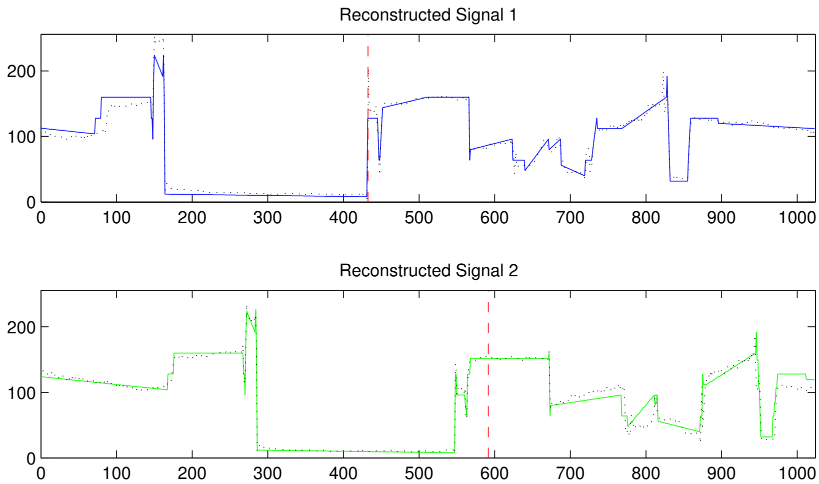

Figure 13: Reconstructed scan lines of stereo images (Figure 12) using a piecewise linear model for the binary tree decomposition and a symmetric distributed compression approach.

4.2 Distributed compression based on the wavelet transform

The wavelet transform has had a tremendous impact on image compression recently and the new image compression standard (Jpeg2000) is based on wavelets. It is therefore natural to explore possible extensions of this transform to the distributed case.

The standard centralized wavelet transform simply consists of two 1-D wavelets applied along the rows and columns of the image. A block diagram of the standard wavelet transform is illustrated in Figure 14. The filter L_0 is usually a low pass filter while H_0 is high pass. Downsampling is first performed on columns and then on rows. The process is usually iterated on the 'low-low' pass version of the image. The resulting transformed image after three iterations is shown in Figure 15. In a classical compression algorithm, the wavelet coefficients are then quantized and entropy encoded.

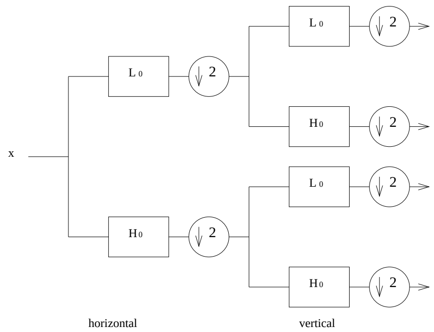

Figure 14: Separable wavelet transform: block diagram

Figure 15: Three iterations of the separable wavelet transform

Our distributed algorithm is based on the geometrical assumptions of Section 3. In particular, we assume that the disparity Δ between the coordinates of the same object in two consecutive images is bounded. Because of this assumption, if we consider two images obtained from two neighbor sensors and take a wavelet transform of each image, we can prove that the two transformed images differ in the high pass components only. Therefore by transmitting a compressed version of one of the two images and only the high pass version of the other one, we can infer, at the decoder, the disparity between the objects in the two images and reconstruct both images faithfully.

More precisely the algorithm operates as follows. The first image is compressed using a classical wavelet based image compression algorithm, the second image is wavelet transformed and only its high-pass components are compressed and transmitted to the receiver. The decoder reconstructs the first image and a high-pass version of the second one. Using a classical disparity block-matching algorithm, the decoder then estimates the disparity between the objects in the two images and reconstruct a more faithful version of the second one.

Simulation results are shown in Figure 16. In this simulation we have not performed quantization of the wavelet coefficients, but simply removed all the wavelet coefficients below a fixed threshold. The threshold was chosen so that only 20% of the coefficients were retained. Figure 16(a) shows the two original stereo images. The compression results for the case in which a classical separate encoder was used, are shown in Figure 16(b). In this case, the average PSNR is 24.9dB. In Figure 16(c) we present the result for our approach where with the same compression rate we can achieve a PSNR of 29.6dB.

5 Conclusions

We have proposed a distributed compression approach for camera sensor networks. In particular, we have shown how simple geometrical information about the scene and the position of the cameras can be used to estimate the correlation structure between different views. Our approach allows for a flexible distribution of the bit-rates amongst the encoders, and can be made resilient to a fixed number of occlusions. Two different approaches to deal with real multi-view images have been also proposed. The first one is based on a quad-tree decomposition of images, while the second one is based on extensions of the wavelet transform. Both methods show good results when applied to real stereo images. For the case of quad-tree algorithm we are at the moment only able to operate on a single scan-line per time. The wavelet methods, on the other hand, operates on the entire image.

References

[1] A. Aaron and B. Girod. Compression with side information using turbo codes. In *IEEE International Conference on Data Compression*, pages 252–261, April 2002.

[2] A. Aaron, P. Ramanathan, and B. Girod. Wyner-Ziv coding of light fields for random access. In *IEEE International Workshop on Multimedia Signal Processing (MMSP)*, September 2004.

[3] E.H. Adelson and J.R. Bergen. The plenoptic function and the elements of early vision. In M. Landy and J. Anthony Movshon, editors, *Computational Models of Visual Processing*, pages 3–20. MIT Press, Cambridge, MA, 1991.

[4] I.F. Akyildiz, Su Weilian, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. *IEEE Communications Magazine*, 40(8):102–114, August 2002.

[5] J.-X. Chai, S.-C. Chan, H.-Y. Shum, and X. Tong. Plenoptic sampling. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 307–318. ACM Press/Addison-Wesley Publishing Co., 2000.

[6] T. P. Coleman, A. H. Lee, M. Medard, and M. Effros. On some new approaches to practical Slepian-Wolf compression inspired by channel coding. In *Data Compression Conference (DCC 2004)*, March 2004.

[7] J. Garcia-Frias. Compression of correlated binary sources using turbo codes. *IEEE Communications Letters*, 5(10):417–419, October 2001.

[8] N. Gehrig and P.L. Dragotti. DIFFERENT-Distributed and Fully Flexible image EncodeRs for camEra sensor NeTworks. In *Proc. of IEEE Int. Conf. on Image Processing(ICIP)*, Genova (Italy), September 2005.

[9] N. Gehrig and P.L. Dragotti. Distributed sampling and compression of scenes with finite rate of innovation in camera sensor networks. In *Proc. of Data Compression Conference (to appear)*, Snowbird, Utah (USA), March 2006.

[10] B. Girod, A. Aaron, S. Rane, and D. Rebollo-Monedero. Distributed video coding. *Proceedings of the IEEE*, 93(1):71–83, January 2005. Invited Paper.

[11] S.J. Gortler, R. Grzeszczuk, R. Szeliski, and M.F. Cohen. The lumigraph. In *SIGGRAPH, Computer Graphics Proceedings*, pages 43–54. ACM Press, 1996.

[12] A. Jagmohan, A. Sehgal, and N. Ahuja. Compression of lightfield rendered images using coset codes. In *Asilomar Conf. on Signals and Systems, Special Session on Distributed Coding*, 2003.

[13] M. Levoy and P. Hanrahan. Light field rendering. In *SIGGRAPH, Computer Graphics Proceedings*, pages 31–40. ACM Press, 1996.

[14] A.D. Liveris, Z. Xiong, and C.N. Georghiades. Compression of Binary Sources With Side Information at the Decoder Using LDPC Codes. *IEEE Communications Letters*, 6(10):440–442, October 2002.

[15] A.D. Liveris, Zixiang Xiong, and C.N. Georghiades. Joint source-channel coding of binary sources with side information at the decoder using IRA codes. In *IEEE Workshop on Multimedia Signal Processing*, pages 53–56, December 2002.

[16] S.S. Pradhan and K. Ramchandran. Distributed source coding using syndromes (DISCUS): Design and construction. In *Data Compression Conference*, pages 158–167, 1999.

[17] S.S. Pradhan and K. Ramchandran. Distributed source coding using syndromes (DISCUS): Design and construction. *IEEE Trans. Information Theory*, 48(3):626–643, March 2003.

[18] R. Puri and K. Ramchandran. PRISM: A video coding architecture based on distributed compression principles. Technical report, EECS Department, University of California, Berkeley, October 2002. article presented in Proc. Of 40thAllerton Conference on Communication, Control, and Computing, Allerton, IL, Oct. 2002.

[19] S. Rane and B. Girod. Analysis of error-resilient video transmission based on systematic source-channel coding. In *Picture Coding Symposium (PCS)*, December 2004.

[20] D. Schonberg, S. S. Pradhan, and K. Ramchandran. Distributed code constructions for the entire Slepian-Wolf rate region for arbitrarily correlated sources. In *Data Compression Conference (DCC 2004)*, March 2004.

[21] A. Sehgal and N. Ahuja. Robust predictive coding and the Wyner-Ziv problem. In *Data Compression Conference*, pages 103–112, March 2003.

[22] A. Sehgal, A. Jagmohan, and N. Ahuja. Wyner-Ziv coding of video: An error-resilient compression framework. *IEEE Trans. on Multimedia*, 6(2):249–258, April 2004.

[23] S. Shamai, S. Verdu, and R. Zamir. Systematic lossy source/channel coding. *IEEE Transactions on Information Theory*, 44(2):564–579, March 1998.

[24] R. Shukla, P.L. Dragotti, M.N. Do, and M. Vetterli. Rate-distortion optimized tree structured compression algorithms for piecewise polynomial images. *IEEE Transactions on Image Processing*, 14(3):343–359, March 2005.

[25] D. Slepian and J.K. Wolf. Noiseless coding of correlated information sources. *IEEE Transactions on Information Theory*, 19(4):471–480, Jul 1973.

[26] V. Stankovic, A.D. Liveris, Z. Xiong, and C.N. Georghiades. Design of Slepian-Wolf codes by channel code partitioning. In *Data Compression Conference (DCC 2004)*, March 2004.

[27] A.D. Wyner. Recent results in the shannon theory. *IEEE Transactions on Information Theory*, 20(1), January 1974.

[28] A.D. Wyner and J. Ziv. The rate-distortion function for source coding with side information at the decoder. *IEEE Transactions on Information Theory*, 22(1):1–10, January 1976.

[29] Z. Xiong, A.D. Liveris, and S.Cheng. Distributed source coding for sensor networks. *IEEE Signal Processing Magazine*, 21(5):80–94, September 2004.

[30] X. Zhu, A. Aaron, and B. Girod. Distributed compression for large camera arrays. In *Proceedings of the IEEE Workshop on Statistical Signal Processing, SSP-2003*, September 2003.

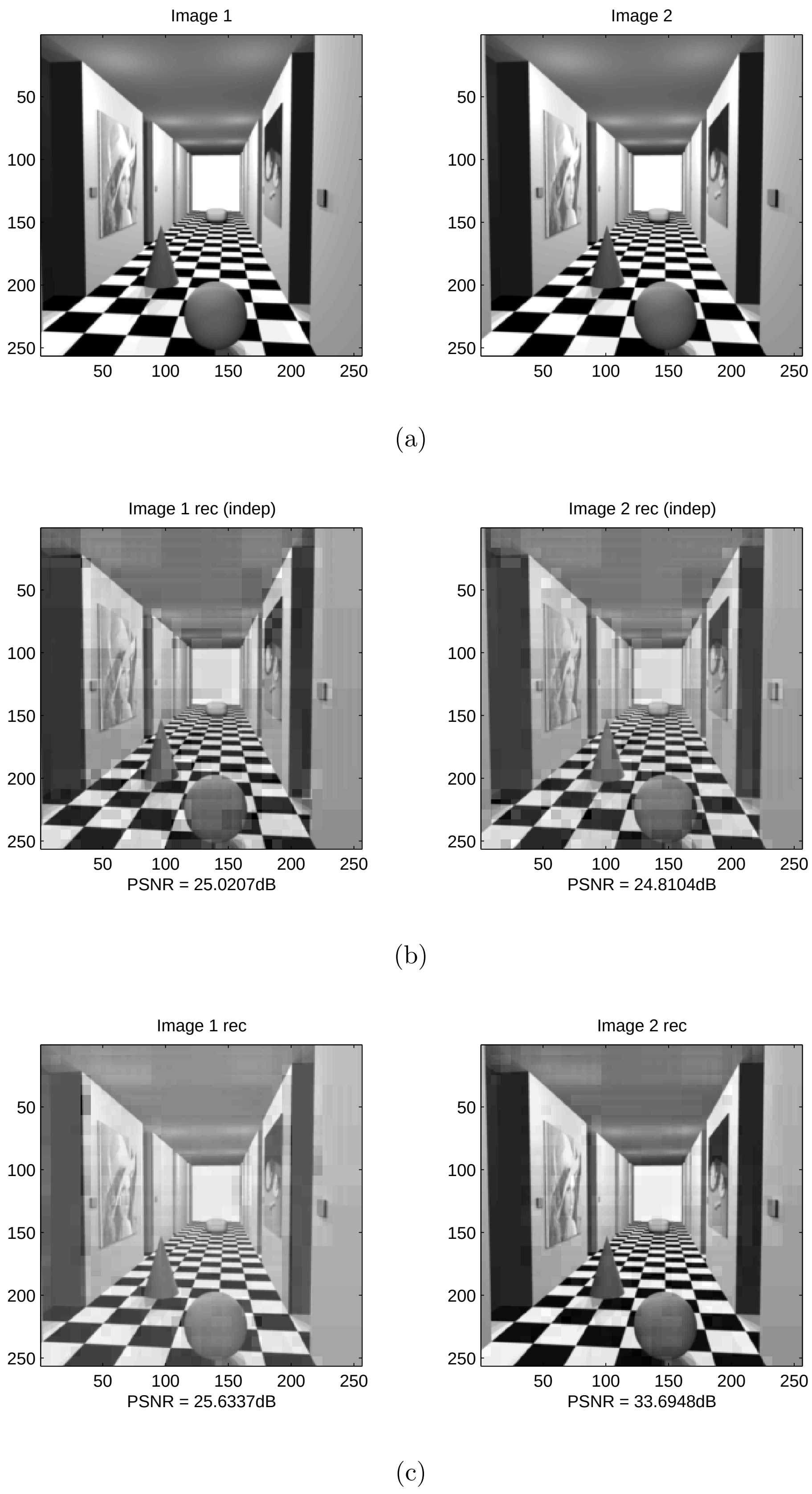

Figure 16: Distributed compression using the wavelet transform. (a) Original stereo images. (b) Separate compression (PSNR=24.9dB). (c) Our Distributed compression algorithm (PSNR=29.6 dB).